CLINTON MacHuki

23/24 Internet Programming (CSYM019-SPR) STUDENT ID: 23852229 

CODE FOR MY premier league WEBSITE TASK 2

# CONTENTS OF MY DATABASE

**Tables**

* teams
* players
* admin \_users
* fixtures
* goals

**Attributes**

* Team
  + TeamID (int)
  + TeamName (varchar (100))
  + TeamLogo (varchar (100))
* Player
  + PlayerID (int)
  + TeamID (int) - foreign key referencing Team table
  + PlayerName (varchar(100))

A screenshot of a computer

Description automatically generated

* Admin User
  + UserID (int)
  + Username (varchar(50))
  + Password (varchar(255))
* Fixture
  + FixtureID (int)
  + HomeTeamID (int) - foreign key referencing Team table
  + AwayTeamID (int) - foreign key referencing Team table
  + Date (date)
  + Result (varchar(100))
* Goal
  + GoalID (int)
  + FixtureID (int) - foreign key referencing Fixture table
  + PlayerID (int) - foreign key referencing Player table
  + Goals (int)

Figure 1 My database

Contents

[CONTENTS OF MY DATABASE 1](#_Toc166974944)

[Application Testing 3](#_Toc166974945)

[CONNECTOR.PHP 7](#_Toc166974946)

[ADD\_PLAYERS.PHP 7](#_Toc166974947)

[ADD\_RESULTS.PHP 11](#_Toc166974948)

[ADD\_SCORERS.PHP 14](#_Toc166974949)

[DASHBOARD.PHP 18](#_Toc166974950)

[DISPLAY\_FIXTURES.PHP 19](#_Toc166974951)

[FIXTURES\_OUTCOME.PHP 23](#_Toc166974952)

[FIXTURES.PHP 26](#_Toc166974953)

[LEAGUE\_TABLE.PHP 30](#_Toc166974954)

[LOGIN.HTML 35](#_Toc166974955)

[LOGIN.PHP 36](#_Toc166974956)

[MANAGEUSERS.PHP 37](#_Toc166974957)

[PLAYER\_FIXTURES.PHP 40](#_Toc166974958)

[PROCESS\_RESULTS.PHP 44](#_Toc166974959)

[PROCESS\_SCORERS.PHP 45](#_Toc166974960)

[REGISTER\_ADMINS.HTML 46](#_Toc166974961)

[REGISTER\_ADMIN.PHP 47](#_Toc166974962)

[STATISTICS.PHP 48](#_Toc166974963)

[TEAMS\_FIXTURES.PHP 57](#_Toc166974964)

[TEAMS\_INPUT.PHP 60](#_Toc166974965)

[TOP\_SCORERS.PHP 62](#_Toc166974966)

[STYLES.CSS 67](#_Toc166974967)

**TO LOGIN USE USERNAME  
username: admin1  
password: admin1**

# Application Testing

To ensure the quality and reliability of the application, I performed a comprehensive series of tests, focusing on the following areas:

**Unit Testing**

I started by testing each PHP function and class in isolation. This helped me verify that individual components of the code were functioning correctly. For instance, I tested functions responsible for calculating league points, updating player statistics, and fetching data from the database. By isolating these functions, I was able to pinpoint any issues and fix them before moving on to more complex tests.

**Integration Testing**

Next, I checked how different modules of the application interact with each other. This involved ensuring that the database connection was correctly established and that data retrieved from the database was accurately processed and displayed. I tested the interaction between the **add\_results.php** script and the league table update mechanism to ensure that new results were correctly reflected in the standings.

**Functional Testing**

I conducted functional testing by evaluating the application against its functional requirements. This included verifying that every feature worked as expected. For example, I:

1. Registered a new user/admin and verified successful registration.
2. Logged in with the newly created account to ensure the login process was secure and functional.
3. Added new teams and players through the respective forms and confirmed they were correctly listed in the database.
4. Added fixtures and results, then checked that the league table and player statistics were updated accordingly.

**End-to-End Testing**

To simulate a real user workflow, I performed end-to-end testing, which involved:

1. Logging in as an admin.
2. Adding a new team and several players.
3. Creating fixtures and entering results.
4. Viewing the updated league table to ensure the changes were accurately reflected.
5. Navigating through different sections of the site, such as the top scorers list and statistics page, to verify that all data was consistent and up-to-date.

**Database Testing**

Ensuring data integrity and validity was crucial. I checked all CRUD operations (Create, Read, Update, Delete) to make sure they were functioning correctly. This included:

1. Adding new records (teams, players, fixtures).
2. Reading and displaying these records accurately on the web pages.
3. Updating records, such as modifying fixture results or player details.
4. Deleting records and ensuring they were removed from the database without affecting related data.

**Security Testing**

I conducted security testing to identify and mitigate vulnerabilities. This involved:

1. Testing for SQL injection by attempting to insert malicious code into input fields.
2. Verifying that user authentication and authorization mechanisms were in place to protect sensitive data and functionalities.
3. Testing if somebody could access the pages that are used for data entry without login in

**Example Test Case for Adding a Fixture**

**Test**

**Description:** Verify that a fixture can be added successfully.

**Pre-conditions:** Admin is logged in.

**Steps:**

1. I navigated to **fixtures.php**.
2. I filled in the fixture details home team vs away team and the date I left the results blank.
3. I submitted the form.

**Expected Result:**

* The fixture was added to the database.
* A success message was displayed.
* The new fixture appeared in the list of display\_fixture.php page.

# CONNECTOR.PHP

I used this code to initiate the code with my database  
username: root  
password:

<?php

// Database connection details

$host = 'localhost'; // Hostname of the database server

$username = 'root'; // Username to connect to the database

$password = ''; // Password to connect to the database

///these are default logins to sql

$database = 'premier\_league'; // Name of the database

// Connect to the database

$conn = new *mysqli*($host, $username, $password, $database); // Create a new MySQLi object to establish a connection

// This line creates a new MySQLi object which represents a connection to the MySQL database. It takes four parameters: hostname, username, password, and database name.

// Check connection

if ($conn->connect\_error) { // Check if there is an error in establishing the connection

    die("Connection failed: " . $conn->connect\_error); // If there is an error, terminate the script and display an error message

}

?>

# ADD\_PLAYERS.PHP

I used this page to enter new players to the database. It has a dropdown to select the team the player belongs to.

<?php

// Include the connector file to establish a database connection

include 'connector.php';

session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

// Define variables and initialize with empty values

$name = $team = ""; // Initialize name and team variables

$name\_err = $team\_err = ""; // Initialize error variables for name and team

// Query to retrieve teams from the database

$sql\_teams = "SELECT TeamID, TeamName FROM Teams"; // SQL query to select TeamID and TeamName from Teams table

$result\_teams = $conn->query($sql\_teams); // Execute the query and store the result in $result\_teams variable

// Processing form data when form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { // Check if the form has been submitted using POST method

    // Validate name

    if (empty(trim($\_POST["name"]))) { // Check if the name field is empty

        $name\_err = "Please enter the player's name."; // Set an error message if name field is empty

    } else {

        $name = trim($\_POST["name"]); // Store the trimmed value of name field in $name variable

    }

    // Validate team

    if (empty(trim($\_POST["team"]))) { // Check if the team field is empty

        $team\_err = "Please select the player's team."; // Set an error message if team field is empty

    } else {

        $team = trim($\_POST["team"]); // Store the trimmed value of team field in $team variable

    }

    // Check input errors before inserting into database

    if (empty($name\_err) && empty($team\_err)) { // Check if there are no errors

        // Prepare an insert statement

        $sql = "INSERT INTO Players (PlayerName, TeamID) VALUES (?, ?)"; // SQL query to insert data into Players table

        if ($stmt = $conn->prepare($sql)) { // Prepare the SQL statement

            // Bind variables to the prepared statement as parameters

            $stmt->bind\_param("si", $param\_name, $param\_team); // Bind parameters

            // Set parameters

            $param\_name = $name; // Set parameter for name

            $param\_team = $team; // Set parameter for team

            // Attempt to execute the prepared statement

            if ($stmt->execute()) { // Execute the prepared statement

                // Redirect to players page

                header("location: add\_player.php"); // Redirect to add\_player.php page

                exit(); // Exit script

            } else {

                echo "Something went wrong. Please try again later."; // Display error message if execution fails

            }

            // Close statement

            $stmt->close(); // Close the prepared statement

        }

    }

    // Close connection

    $conn->close(); // Close the database connection

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Add New Player</title>

    <link rel="stylesheet" href="styles.css"> <!-- Include stylesheet -->

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Add New Player</h2>

    <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="post">

        <div>

            <label>Name</label> <!-- Input field for player's name -->

            <input type="text" name="name" value="<?php echo $name; ?>"> <!-- Display entered name if form submission fails -->

            <span><?php echo $name\_err; ?></span> <!-- Display name error message -->

        </div>

        <div>

            <label>Team</label> <!-- Dropdown for selecting player's team -->

            <select name="team">

                <option value="">Select a team</option> <!-- Default option -->

                <?php

                // Loop through teams data and display options

                if ($result\_teams->num\_rows > 0) { // Check if there are teams in the database

                    while ($row = $result\_teams->fetch\_assoc()) { // Iterate through each team

                        echo "<option value='" . $row["TeamID"] . "'>" . $row["TeamName"] . "</option>"; // Display team options

                    }

                }

                ?>

            </select>

            <span><?php echo $team\_err; ?></span> <!-- Display team error message -->

        </div>

        <div>

            <input type="submit" value="Submit"> <!-- Submit button -->

            <input type="reset" value="Reset"> <!-- Reset button to clear form -->

        </div>

    </form>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# ADD\_RESULTS.PHP

This page is used to add results to fixtures that are added to the database without results….i.e if a fixture exists in the database and the result hasn’t been posted you can click a dropdown and select the fixture and add the results

<?php

// Include the connector file to establish a database connection

include 'connector.php';

session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

// Fetch fixtures from the database

$sql = "SELECT FixtureID, HomeTeamID, AwayTeamID, Result, *Date* FROM Fixtures"; // SQL query to select fixtures from Fixtures table

$result = $conn->query($sql); // Execute the query and store the result

$fixtures = array(); // Initialize an empty array to store fixtures

if ($result->num\_rows > 0) { // Check if there are fixtures in the result

    while($row = $result->fetch\_assoc()) { // Loop through each row in the result

        $fixtures[] = $row; // Add the row to the fixtures array

    }

}

// Fetch team names from the Teams table

$sql = "SELECT TeamID, TeamName FROM Teams"; // SQL query to select TeamID and TeamName from Teams table

$result = $conn->query($sql); // Execute the query and store the result

$teams = array(); // Initialize an empty array to store team names

if ($result->num\_rows > 0) { // Check if there are teams in the result

    while($row = $result->fetch\_assoc()) { // Loop through each row in the result

        $teams[$row['TeamID']] = $row['TeamName']; // Store team names with their corresponding IDs

    }

}

// Close the database connection

$conn->close(); // Close the database connection

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Add Results to Fixtures</title>

    <link rel="stylesheet" href="styles.css"> <!-- Include stylesheet -->

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Add Results to Fixtures</h2>

    <form action="process\_results.php" method="POST"> <!-- Form to add results -->

        <label for="fixture">Select Fixture:</label>

        <select id="fixture" name="fixture" required> <!-- Dropdown to select fixture -->

            <option value="">Select Fixture</option> <!-- Default option -->

            <?php foreach ($fixtures as $fixture): ?> <!-- Iterate through fixtures array -->

                <?php if(empty($fixture['Result'])): ?> <!-- Check if result is empty -->

                    <option value="<?php echo $fixture['FixtureID']; ?>"> <!-- Display fixture option -->

                        <?php echo $teams[$fixture['HomeTeamID']] . ' vs ' . $teams[$fixture['AwayTeamID']] . ' - ' . $fixture['Date']; ?> <!-- Display fixture details -->

                    </option>

                <?php endif; ?>

            <?php endforeach; ?>

        </select><br><br>

        <label for="result">Result:</label>

        <input type="text" id="result" name="result" required><br><br> <!-- Input field for result -->

        <button type="submit">Add Result</button> <!-- Submit button -->

    </form>

    <script>

        // Function to show only fixtures where no results have been posted yet

*function* filterFixtures() {

*var* fixtures = document.getElementById("fixture").getElementsByTagName("option"); // Get all fixture options

            for (*var* i = 0; i < fixtures.length; i++) { // Loop through each option

                if (fixtures[i].value !== "" && fixtures[i].textContent.includes(" - ")) { // Check if option is valid

                    fixtures[i].style.display = "block"; // Show the fixture option

                } else {

                    fixtures[i].style.display = "none"; // Hide the fixture option

                }

            }

        }

        // Call the function when the page loads and when the user changes the result input

        window.onload = filterFixtures; // Call filterFixtures function when the page loads

        document.getElementById("result").addEventListener("change", filterFixtures); // Call filterFixtures function when the result input changes

    </script>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# ADD\_SCORERS.PHP

This page is used to add scorers to a fixture, when a fixture is selected the code filatures the players and only plays associated with the two teams are displayed in the dropdown, the user can then proceed to adding the number of goals they scored in the specific fixture.

<?php

// Include the connector file to establish a database connection

include 'connector.php';

session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

// Fetch fixtures from the database

$sql = "SELECT FixtureID, HomeTeamID, AwayTeamID, *Date* FROM Fixtures"; // SQL query to select fixtures from Fixtures table

$result = $conn->query($sql); // Execute the query and store the result

$fixtures = array(); // Initialize an empty array to store fixtures

if ($result->num\_rows > 0) { // Check if there are fixtures in the result

    while($row = $result->fetch\_assoc()) { // Loop through each row in the result

        $fixtures[] = $row; // Add the row to the fixtures array

    }

}

// Fetch team names from the Teams table

$sql = "SELECT TeamID, TeamName FROM Teams"; // SQL query to select TeamID and TeamName from Teams table

$result = $conn->query($sql); // Execute the query and store the result

$teams = array(); // Initialize an empty array to store team names

if ($result->num\_rows > 0) { // Check if there are teams in the result

    while($row = $result->fetch\_assoc()) { // Loop through each row in the result

        $teams[$row['TeamID']] = $row['TeamName']; // Store team names with their corresponding IDs

    }

}

// Fetch players from the Players table

$sql = "SELECT PlayerID, PlayerName, TeamID FROM Players"; // SQL query to select PlayerID, PlayerName, and TeamID from Players table

$result = $conn->query($sql); // Execute the query and store the result

$players = array(); // Initialize an empty array to store players

if ($result->num\_rows > 0) { // Check if there are players in the result

    while($row = $result->fetch\_assoc()) { // Loop through each row in the result

        $players[$row['PlayerID']] = array( // Store player details in the players array

            'PlayerName' => $row['PlayerName'], // Store player name

            'TeamID' => $row['TeamID'] // Store team ID

        );

    }

}

// Close the database connection

$conn->close(); // Close the database connection

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Add Scorers</title>

    <link rel="stylesheet" href="styles.css"> <!-- Include stylesheet -->

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Add Scorers</h2>

    <form action="process\_scorers.php" method="POST" id="scorersForm"> <!-- Form to add scorers -->

        <label for="fixture">Select Fixture:</label>

        <select id="fixture" name="fixture" required> <!-- Dropdown to select fixture -->

            <option value="">Select Fixture</option> <!-- Default option -->

            <?php foreach ($fixtures as $fixture): ?> <!-- Iterate through fixtures array -->

                <option value="<?php echo $fixture['FixtureID']; ?>"> <!-- Display fixture option -->

                    <?php echo $teams[$fixture['HomeTeamID']] . ' vs ' . $teams[$fixture['AwayTeamID']] . ' - ' . $fixture['Date']; ?> <!-- Display fixture details -->

                </option>

            <?php endforeach; ?>

        </select><br><br>

        <div id="scorersContainer">

            <div class="scorerEntry">

                <label for="player1">Select Scorer:</label>

                <select name="player[]" class="playerSelect" required> <!-- Dropdown to select player -->

                    <option value="">Select Scorer</option> <!-- Default option -->

                    <?php foreach ($players as $playerID => $playerData): ?> <!-- Iterate through players array -->

                        <option value="<?php echo $playerID; ?>"><?php echo $playerData['PlayerName']; ?></option> <!-- Display player options -->

                    <?php endforeach; ?>

                </select>

                <label for="goals1">Goals:</label>

                <input type="number" name="goals[]" class="goalInput" min="1" required> <!-- Input field for goals -->

            </div>

        </div>

        <button type="button" id="addScorerButton">Add Scorer</button> <!-- Button to add more scorers -->

        <br><br>

        <button type="submit">Add Scorers</button> <!-- Submit button -->

    </form>

    <script>

    document.getElementById("fixture").addEventListener("change", *function*() { // Add an event listener to the fixture dropdown for change event

*var* fixtureId = this.value; // Get the selected fixture ID

*var* selectedHomeTeamId = <?php echo json\_encode($fixtures[0]['HomeTeamID']); ?>; // Set default selected home team ID from PHP variable

*var* selectedAwayTeamId = <?php echo json\_encode($fixtures[0]['AwayTeamID']); ?>; // Set default selected away team ID from PHP variable

        // Find selected home and away team IDs based on the selected fixture ID

        <?php foreach ($fixtures as $fixture): ?> // Iterate through fixtures array

            if (fixtureId == <?php echo $fixture['FixtureID']; ?>) { // Check if the fixture ID matches the selected fixture

                selectedHomeTeamId = <?php echo $fixture['HomeTeamID']; ?>; // Set the selected home team ID

                selectedAwayTeamId = <?php echo $fixture['AwayTeamID']; ?>; // Set the selected away team ID

            }

        <?php endforeach; ?>

        // Filter players based on the selected teams

*var* players = <?php echo json\_encode($players); ?>; // Get players data from PHP variable

*var* playerSelects = document.getElementsByClassName("playerSelect"); // Get all elements with class 'playerSelect'

        Array.from(playerSelects).forEach(*function*(*select*) { // Iterate through each player select element

*var* teamId = select.dataset.teamId; // Get the team ID associated with the select element

            select.innerHTML = ""; // Clear the current options of the select element

            select.appendChild(document.createElement("option")); // Add default option to the select element

            Object.entries(players).forEach(([*playerId*, *playerData*]) *=>* { // Iterate through each player data

*var* playerTeamId = playerData['TeamID']; // Get the team ID of the player

                if (playerTeamId == selectedHomeTeamId || playerTeamId == selectedAwayTeamId) { // Check if player belongs to selected home or away team

*var* option = document.createElement("option"); // Create a new option element

                    option.value = playerId; // Set the value of the option to player ID

                    option.textContent = playerData['PlayerName']; // Set the text content of the option to player name

                    select.appendChild(option); // Append the option to the select element

                }

            });

        });

    });

</script>

<footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# DASHBOARD.PHP

This is the admin landing page. When the admin logs in they land on this page displaying all the available functionalities in the website.

<?php

session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Dashboard</title>

    <link rel="stylesheet" href="styles.css"> <!-- Link to external CSS file -->

</head>

<body>

<h2>Dashboard</h2>

<div class="container">

<ul class="dashboard-grid"> <!-- Add a class to the ul element for styling -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

        <li><a href="manageusers.php">Manage Admin</a></li>

    </ul>

</div>

</body>

<footer>

    <p>&copy; 2024 EPL. All rights reserved.</p>

</footer>

</html>

# DISPLAY\_FIXTURES.PHP

This page displays the fixtures available in the database. The fixtures are combined and separated depending on the data played.

<?php

include 'connector.php'; // Include the database connector file

// Fetch fixtures from the database

$sql = "SELECT f.FixtureID, h.TeamName AS HomeTeam, a.TeamName AS AwayTeam, f.Date, f.Result

        FROM Fixtures f

        INNER JOIN Teams h ON f.HomeTeamID = h.TeamID

        INNER JOIN Teams a ON f.AwayTeamID = a.TeamID

        ORDER BY f.Date DESC"; // SQL query to fetch fixtures with home and away team names, ordered by date in descending order

$result = $conn->query($sql); // Execute the query

$fixtures = array(); // Initialize an empty array to store fixtures data

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        $fixtures[] = $row; // Add each row (fixture data) to the fixtures array

    }

}

// Close the database connection

$conn->close(); // Close the database connection after fetching data

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Display Fixtures</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

<ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

    <h2>Fixtures</h2>

    <div class="table-container">

    <?php

    // Initialize variable to store previous date

    $prevDate = null;

    // Loop through fixtures to display them grouped by date

    foreach ($fixtures as $fixture) {

        // Get the date of the current fixture

        $currDate = $fixture['Date'];

        // Check if it's a new date

        if ($currDate !== $prevDate) {

            // If it's a new date, close the previous table if it exists

            if ($prevDate !== null) {

                echo '</table>';

            }

            // Display the date as a heading

            echo '<h3>' . $currDate . '</h3>';

            // Open a new table for fixtures of the current date

            echo '<table border="1" cellpadding="5" cellspacing="0">

                <tr>

                    <th>Home Team</th>

                    <th>Result</th>

                    <th>Away Team</th>

                </tr>';

        }

        // Display the fixture row

        echo '<tr class="clickable-row" data-href="fixture\_outcome.php?fixture\_id=' . $fixture['FixtureID'] . '">

            <td>' . $fixture['HomeTeam'] . '</td>

            <td>' . $fixture['Result'] . '</td>

            <td>' . $fixture['AwayTeam'] . '</td>

        </tr>';

        // Update the previous date

        $prevDate = $currDate;

    }

    // Close the last table

    echo '</table>';

    ?>

    <!-- JavaScript to make rows clickable -->

    <script>

        document.addEventListener("DOMContentLoaded", *function*() {

*var* rows = document.querySelectorAll(".clickable-row");

            rows.forEach(*function*(*row*) {

                row.addEventListener("click", *function*() {

*var* href = row.getAttribute("data-href");

                    window.location.href = href;

                });

            });

        });

    </script>

    </div>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

    <script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

</body>

</html>

# FIXTURES\_OUTCOME.PHP

This page can be accessed by selectin a fixture in the fixture\_display.php page. It contains more information about a fixture including the scorers of the fixture.

<?php

include 'connector.php'; // Include the database connector file

// Get the fixture ID from the URL parameter

if (isset($\_GET['fixture\_id'])) {

    $fixture\_id = $\_GET['fixture\_id'];

} else {

    // Redirect to display\_fixtures.php if fixture\_id parameter is not provided

    header("Location: display\_fixtures.php");

    exit();

}

// Fetch fixture outcome from the database based on the provided fixture ID

$sql = "SELECT f.FixtureID, h.TeamName AS HomeTeam, a.TeamName AS AwayTeam, f.Date, f.Result

        FROM Fixtures f

        INNER JOIN Teams h ON f.HomeTeamID = h.TeamID

        INNER JOIN Teams a ON f.AwayTeamID = a.TeamID

        WHERE f.FixtureID = $fixture\_id"; // SQL query to fetch fixture outcome for the provided fixture ID

$result = $conn->query($sql); // Execute the query

$fixture = $result->fetch\_assoc(); // Fetch the fixture data

// Fetch goal scorers for the fixture from the database

$sql = "SELECT g.PlayerID, g.goals, p.PlayerName, t.TeamName

        FROM goals g

        INNER JOIN Players p ON g.PlayerID = p.PlayerID

        INNER JOIN Teams t ON p.TeamID = t.TeamID

        WHERE g.FixtureID = $fixture\_id"; // SQL query to fetch goal scorers for the provided fixture ID

$result = $conn->query($sql); // Execute the query

$goal\_scorers = array(); // Initialize an empty array to store goal scorers data

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        $goal\_scorers[] = $row; // Add each row (goal scorer data) to the goal\_scorers array

    }

}

// Close the database connection

$conn->close(); // Close the database connection after fetching data

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Fixture Outcome</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <div class="container">

    <ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

        <h2 class="heading">Fixture Outcome</h2>

        <div class="fixture-details">

            <h3>Fixture Details</h3>

            <p><strong>Home Team:</strong> <?php echo $fixture['HomeTeam']; ?></p>

            <p><strong>Away Team:</strong> <?php echo $fixture['AwayTeam']; ?></p>

            <p><strong>Date:</strong> <?php echo $fixture['Date']; ?></p>

            <p><strong>Result:</strong> <?php echo $fixture['Result']; ?></p>

        </div>

        <div class="goal-scorers">

            <h3>Goal Scorers</h3>

            <ul class="scorers-list">

                <?php foreach ($goal\_scorers as $scorer): ?>

                    <li><?php echo $scorer['PlayerName']; ?> (<?php echo $scorer['TeamName']; ?>) - <?php echo $scorer['goals']; ?> goals</li> <!-- Display goal scorers, their teams, and the number of goals -->

                <?php endforeach; ?>

            </ul>

        </div>

    </div>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

    <script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

</body>

</html>

# FIXTURES.PHP

This page is used to add new fixtures to the database. A user csan add a fixture with or without the results. This was important when developing the site for easy data input. You can add fixtures without results and when the results are out you can update the results using page add\_results.php

<?php

include 'connector.php'; // Include the database connector file

session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { // Check if the request method is POST

    // Retrieve form data

    $home\_team\_id = $\_POST['home\_team']; // Retrieve home team ID from the form

    $away\_team\_id = $\_POST['away\_team']; // Retrieve away team ID from the form

    $date = $\_POST['date']; // Retrieve date from the form

    $result = $\_POST['result']; // Retrieve result from the form (assuming it can be a string)

    // Insert new fixture into the database

    $sql = "INSERT INTO Fixtures (HomeTeamID, AwayTeamID, *Date*, Result) VALUES ('$home\_team\_id', '$away\_team\_id', '$date', '$result')";

    if ($conn->query($sql) === TRUE) { // Execute the SQL query and check if it was successful

        // Redirect to fixtures.php with success message

        header("Location: fixtures.php?status=success"); // Redirect to fixtures.php with a success status in the URL

        exit(); // Exit the script

    } else {

        // Redirect to fixtures.php with error message

        header("Location: fixtures.php?status=error"); // Redirect to fixtures.php with an error status in the URL

        exit(); // Exit the script

    }

}

// Fetch team names from the Teams table

$sql = "SELECT TeamID, TeamName FROM Teams"; // SQL query to fetch team names from the Teams table

$result = $conn->query($sql); // Execute the SQL query

$teams = array(); // Initialize an empty array to store team names

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        $teams[] = $row; // Add each row (team data) to the teams array

    }

}

// Close the database connection

$conn->close(); // Close the database connection after fetching data

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Fixtures</title> <!-- Title of the webpage -->

    <link rel="stylesheet" href="styles.css"> <!-- Link to an external stylesheet -->

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Fixtures</h2> <!-- Heading for the page -->

    <?php

    // Display status message if provided in the URL

    if (isset($\_GET['status'])) { // Check if 'status' parameter is set in the URL

        if ($\_GET['status'] == 'success') { // Check if the status is 'success'

            echo '<p style="color: green;">Fixture added successfully!</p>'; // Display success message

        } elseif ($\_GET['status'] == 'error') { // Check if the status is 'error'

            echo '<p style="color: red;">Error adding fixture. Please try again!</p>'; // Display error message

        }

    }

    ?>

    <form action="fixtures.php" method="POST"> <!-- Form to add a new fixture -->

        <label for="home\_team">Home Team:</label> <!-- Label for home team select dropdown -->

        <select id="home\_team" name="home\_team" required> <!-- Select dropdown for home team -->

            <option value="">Select Home Team</option> <!-- Default option for home team -->

            <?php foreach ($teams as $team): ?> <!-- Loop through each team -->

                <option value="<?php echo $team['TeamID']; ?>"><?php echo $team['TeamName']; ?></option> <!-- Option for each team -->

            <?php endforeach; ?>

        </select><br><br>

        <label for="away\_team">Away Team:</label> <!-- Label for away team select dropdown -->

        <select id="away\_team" name="away\_team" required> <!-- Select dropdown for away team -->

            <option value="">Select Away Team</option> <!-- Default option for away team -->

            <?php foreach ($teams as $team): ?> <!-- Loop through each team -->

                <option value="<?php echo $team['TeamID']; ?>"><?php echo $team['TeamName']; ?></option> <!-- Option for each team -->

            <?php endforeach; ?>

        </select><br><br>

        <label for="date">Date:</label> <!-- Label for date input -->

        <input type="date" id="date" name="date" required><br><br> <!-- Input field for date -->

        <label for="result">Result:</label> <!-- Label for result input -->

        <input type="text" id="result" name="result"><br><br> <!-- Input field for result -->

        <button type="submit">Add Fixture</button> <!-- Submit button to add fixture -->

    </form>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# LEAGUE\_TABLE.PHP

This page generates the league table using the data available in the results table in the database. The teams are also clickable to enable user to navigate into a team and see their fixtures and the results of their previous fixtures.

<?php

include 'connector.php'; // Include the database connector file

// Fetch fixtures from the database

$sql = "SELECT FixtureID, HomeTeamID, AwayTeamID, Result FROM Fixtures"; // SQL query to fetch fixtures

$result = $conn->query($sql); // Execute the SQL query

$fixtures = array(); // Initialize an empty array to store fixtures

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        if (!empty($row['Result'])) { // Check if the result is not empty

            $fixtures[] = $row; // Add the fixture to the fixtures array

        }

    }

}

// Initialize arrays for storing team statistics

$points = array();

$played = array();

$won = array();

$drawn = array();

$lost = array();

$goalsFor = array();

$goalsAgainst = array();

$goalDifference = array();

// Fetch team names from the database

$sql = "SELECT TeamID, TeamName FROM Teams"; // SQL query to fetch team names

$result = $conn->query($sql); // Execute the SQL query

$teams = array(); // Initialize an empty array to store teams

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        $teamID = $row['TeamID']; // Get the team ID

        $teamName = $row['TeamName']; // Get the team name

        $teams[$teamID] = $teamName; // Add the team to the teams array

        // Initialize team statistics arrays

        $points[$teamName] = 0;

        $played[$teamName] = 0;

        $won[$teamName] = 0;

        $drawn[$teamName] = 0;

        $lost[$teamName] = 0;

        $goalsFor[$teamName] = 0;

        $goalsAgainst[$teamName] = 0;

        $goalDifference[$teamName] = 0;

    }

}

// Process results and calculate points for each team

foreach ($fixtures as $fixture) {

    $result = $fixture['Result'];

    list($home\_score, $away\_score) = explode('-', $result);

    // Update played matches

    $played[$teams[$fixture['HomeTeamID']]]++;

    $played[$teams[$fixture['AwayTeamID']]]++;

    // Update goals for and against

    $goalsFor[$teams[$fixture['HomeTeamID']]] += $home\_score;

    $goalsFor[$teams[$fixture['AwayTeamID']]] += $away\_score;

    $goalsAgainst[$teams[$fixture['HomeTeamID']]] += $away\_score;

    $goalsAgainst[$teams[$fixture['AwayTeamID']]] += $home\_score;

    if ($home\_score > $away\_score) {

        // Home team wins

        $won[$teams[$fixture['HomeTeamID']]]++;

        $lost[$teams[$fixture['AwayTeamID']]]++;

        $points[$teams[$fixture['HomeTeamID']]] += 3;

    } elseif ($home\_score == $away\_score) {

        // Draw

        $drawn[$teams[$fixture['HomeTeamID']]]++;

        $drawn[$teams[$fixture['AwayTeamID']]]++;

        $points[$teams[$fixture['HomeTeamID']]] += 1;

        $points[$teams[$fixture['AwayTeamID']]] += 1;

    } else {

        // Away team wins

        $won[$teams[$fixture['AwayTeamID']]]++;

        $lost[$teams[$fixture['HomeTeamID']]]++;

        $points[$teams[$fixture['AwayTeamID']]] += 3;

    }

}

// Calculate goal difference

foreach ($teams as $teamID => $teamName) {

    $goalDifference[$teamName] = $goalsFor[$teamName] - $goalsAgainst[$teamName];

}

// Custom sorting function to sort by points and then by goal difference

uasort($teams, *function*($a, $b) use ($points, $goalDifference) {

    if ($points[$a] == $points[$b]) { // Check if points are equal

        // If points are equal, compare goal difference

        return $goalDifference[$b] - $goalDifference[$a];

    }

    // Otherwise, compare points

    return $points[$b] - $points[$a];

});

// Close the database connection

$conn->close(); // Close the database connection after fetching data

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>League Table</title>

    <link rel="stylesheet" href="styles.css"> <!-- Link to an external stylesheet -->

</head>

<body>

    <ul>

    <ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

    </ul>

    <h2>League Table</h2> <!-- Heading for the league table -->

    <div class="table-container">

    <table border="1" id="league-table"> <!-- Table to display the league table -->

        <tr>

            <th>Position</th>

            <th>Team</th>

            <th>Played</th>

            <th>Won</th>

            <th>Drawn</th>

            <th>Lost</th>

            <th>Goals For</th>

            <th>Goals Against</th>

            <th>Goal Difference</th>

            <th>Points</th>

        </tr>

        <?php $position = 1; ?> <!-- Initialize position counter -->

        <?php foreach ($teams as $teamID => $teamName): ?> <!-- Loop through teams -->

            <tr class="clickable-row" data-href="team\_fixtures.php?teamID=<?php echo $teamID; ?>">

                <td><?php echo $position; ?></td> <!-- Display position -->

                <td><?php echo $teamName; ?></td> <!-- Display team name -->

                <td><?php echo $played[$teamName]; ?></td> <!-- Display matches played -->

                <td><?php echo $won[$teamName]; ?></td> <!-- Display matches won -->

                <td><?php echo $drawn[$teamName]; ?></td> <!-- Display matches drawn -->

                <td><?php echo $lost[$teamName]; ?></td> <!-- Display matches lost -->

                <td><?php echo $goalsFor[$teamName]; ?></td> <!-- Display goals scored -->

                <td><?php echo $goalsAgainst[$teamName]; ?></td> <!-- Display goals conceded -->

                <td><?php echo $goalDifference[$teamName]; ?></td> <!-- Display goal difference -->

                <td><?php echo $points[$teamName]; ?></td> <!-- Display points -->

            </tr>

            <?php $position++; ?> <!-- Increment position counter -->

        <?php endforeach; ?> <!-- End of team loop -->

    </table>

        </div>

    <script>

        document.addEventListener("DOMContentLoaded", *function*() {

*var* rows = document.querySelectorAll(".clickable-row"); // Select all rows with the class "clickable-row"

            rows.forEach(*function*(*row*) {

                row.addEventListener("click", *function*() {

*var* href = row.getAttribute("data-href"); // Get the URL from the "data-href" attribute

                    window.location.href = href; // Navigate to the URL when the row is clicked

                });

            });

        });

    </script>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

    <script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

</body>

</html>

# LOGIN.HTML

Page login is used by the administrator to login to the website and be able to manipulate the data in the website.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <h2>Login</h2> <!-- Heading for the login form -->

    <form action="login.php" method="POST"> <!-- Form for submitting login credentials -->

        <label for="username">Username:</label> <!-- Label for the username input -->

        <input type="text" id="username" name="username" required><br><br> <!-- Text input for entering username -->

        <label for="password">Password:</label> <!-- Label for the password input -->

        <input type="password" id="password" name="password" required><br><br> <!-- Password input for entering password -->

        <button type="submit">Login</button> <!-- Button to submit login credentials -->

    </form>

</body>

</html>

# LOGIN.PHP

This pages enables the user to login by accessing the database using php. If success the user id directed to page dashboard.php. It also sets the session cookie username.

<?php

session\_start();

include 'connector.php';

// Handle form submission

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    // Query to fetch user data based on username

    $sql = "SELECT \* FROM admin\_users WHERE username='$username'";

    $result = $conn->query($sql);

    if ($result->num\_rows == 1) {

        // Username exists, verify password

        $row = $result->fetch\_assoc();

        if (password\_verify($password, $row['password'])) {

            // Password is correct, start session, set cookie, and redirect to dashboard

            $\_SESSION['username'] = $username;

            setcookie("username", $username, time() + (86400 \* 30), "/"); // Cookie lasts for 30 days

            header("Location: dashboard.php");

            exit();

        } else {

            // Incorrect password, redirect to login page with error message

            header("Location: login.html?error=1");

            exit();

        }

    } else {

        // Username not found, redirect to login page with error message

        header("Location: login.html?error=2");

        exit();

    }

}

$conn->close();

?>

# MANAGEUSERS.PHP

This page can be used to add new users or allow the administrator to chang the passwords of other administrators.

<?php

session\_start();

include 'connector.php';

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

// Handle form submissions

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Add New User

    if (isset($\_POST['add\_user'])) {

        $new\_username = $\_POST['new\_username'];

        $new\_password = $\_POST['new\_password'];

        // Hash the password

        $hashed\_password = password\_hash($new\_password, PASSWORD\_DEFAULT);

        // Insert new user into the database

        $sql = "INSERT INTO admin\_users (username, password) VALUES (?, ?)";

        $stmt = $conn->prepare($sql);

        $stmt->bind\_param("ss", $new\_username, $hashed\_password);

        $stmt->execute();

    }

    // Edit User Password

    elseif (isset($\_POST['edit\_password'])) {

        $user\_id = $\_POST['user\_id'];

        $new\_password = $\_POST['new\_password'];

        // Hash the new password

        $hashed\_password = password\_hash($new\_password, PASSWORD\_DEFAULT);

        // Update user password in the database

        $sql = "UPDATE admin\_users SET password=? WHERE user\_id=?";

        $stmt = $conn->prepare($sql);

        $stmt->bind\_param("si", $hashed\_password, $user\_id);

        $stmt->execute();

    }

}

// Fetch users from the database

$sql = "SELECT \* FROM admin\_users";

$result = $conn->query($sql);

$users = array(); // Initialize an array to store users

if ($result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        $users[] = $row; // Add each user to the users array

    }

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User Management</title>

    <link rel="stylesheet" href="styles.css"> <!-- Include your CSS file -->

</head>

<body>

<ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Add New User</h2>

    <form action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="POST">

        <input type="text" name="new\_username" placeholder="Username">

        <input type="password" name="new\_password" placeholder="Password">

        <button type="submit" name="add\_user">Add User</button>

    </form>

    <h2>Users</h2>

    <table>

        <tr>

            <th>Username</th>

            <th>Password</th>

            <th>Action</th>

        </tr>

        <?php foreach ($users as $user): ?>

            <tr>

                <td><?php echo $user['username']; ?></td>

                <td><?php echo $user['password']; ?></td>

                <td>

                    <form action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="POST">

                        <input type="hidden" name="user\_id" value="<?php echo $user['user\_id']; ?>">

                        <input type="password" name="new\_password" placeholder="New Password">

                        <button type="submit" name="edit\_password">Edit Password</button>

                    </form>

                </td>

            </tr>

        <?php endforeach; ?>

    </table>

</body>

</html>

# PLAYER\_FIXTURES.PHP

This page can be accessed by clicking on a player in page topscorer. This page displayes the fixtures in which a player scored in.

<?php

include 'connector.php'; // Include the database connector file

// Check if the player ID is provided in the URL

if (isset($\_GET['player\_id'])) {

    $playerId = $\_GET['player\_id'];

    // Fetch player information from the database

    $sql = "SELECT PlayerName FROM Players WHERE PlayerID = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("i", $playerId);

    $stmt->execute();

    $result = $stmt->get\_result();

    if ($result->num\_rows > 0) {

        $player = $result->fetch\_assoc();

        $playerName = $player['PlayerName'];

        // Fetch fixtures in which the player scored from the database

        $sql = "SELECT f.Date, t1.TeamName AS HomeTeam, t2.TeamName AS AwayTeam

                FROM Fixtures f

                INNER JOIN Goals g ON f.FixtureID = g.FixtureID

                INNER JOIN Teams t1 ON f.HomeTeamID = t1.TeamID

                INNER JOIN Teams t2 ON f.AwayTeamID = t2.TeamID

                WHERE g.PlayerID = ?";

        $stmt = $conn->prepare($sql);

        $stmt->bind\_param("i", $playerId);

        $stmt->execute();

        $result = $stmt->get\_result();

        $fixtures = array(); // Initialize an array to store fixtures data

        if ($result->num\_rows > 0) {

            while($row = $result->fetch\_assoc()) {

                $fixtures[] = $row; // Add each row (fixture data) to the fixtures array

            }

        } else {

            echo "No fixtures found for this player.";

        }

    } else {

        echo "Player not found.";

    }

} else {

    echo "Player ID not provided.";

}

// Close the database connection

$stmt->close();

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title><?php echo $playerName; ?>'s</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

    <div class="table-container">

    <h2><?php echo $playerName; ?>'s Has scored in the following fixtures</h2>

    <table border="1" cellpadding="5" cellspacing="0">

    <!-- Table opening tag with border, cell padding, and cell spacing attributes -->

    <tr>

        <!-- Table row opening tag for the table header row -->

        <th>Date</th>

        <!-- Table header cell for the date column -->

        <th>Home Team</th>

        <!-- Table header cell for the home team column -->

        <th>Away Team</th>

        <!-- Table header cell for the away team column -->

    </tr>

    <!-- Table row closing tag for the table header row -->

    <?php foreach ($fixtures as $fixture): ?>

        <!-- PHP foreach loop to iterate over each fixture in the $fixtures array -->

        <tr>

            <!-- Table row opening tag for each fixture -->

            <td><?php echo $fixture['Date']; ?></td>

            <!-- Table data cell containing the date of the fixture -->

            <td><?php echo $fixture['HomeTeam']; ?></td>

            <!-- Table data cell containing the home team name of the fixture -->

            <td><?php echo $fixture['AwayTeam']; ?></td>

            <!-- Table data cell containing the away team name of the fixture -->

        </tr>

        <!-- Table row closing tag for each fixture -->

    <?php endforeach; ?>

    <!-- End of the PHP foreach loop -->

</table>

<!-- Table closing tag -->

    </div>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

    <script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

</body>

</html>

# PROCESS\_RESULTS.PHP

This is the php code that is used to process the results of a fixture before adding them to the database.

<?php

include 'connector.php';

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Retrieve form data

    $fixture\_id = $\_POST['fixture'];

    $result = $\_POST['result']; // Assuming result can be a string

    // Insert result into the database

    $sql = "UPDATE Fixtures SET Result = '$result' WHERE FixtureID = $fixture\_id";

    if ($conn->query($sql) === TRUE) {

        // Redirect to add\_scorers.php with success message

        header("Location: add\_scorers.php");

        exit();

    } else {

        // Redirect to add\_scorers.php with error message

        header("Location: add\_scorers.php");

        exit();

    }

}

// Close the database connection

$conn->close();

?>

# PROCESS\_SCORERS.PHP

This is the php code that is used to process the scorers data before adding it into the database.

<?php

include 'connector.php';

// Check if form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Validate form data

    if (isset($\_POST['fixture']) && isset($\_POST['player']) && isset($\_POST['goals'])) {

        // Retrieve form data

        $fixtureID = $\_POST['fixture'];

        $players = $\_POST['player'];

        $goals = $\_POST['goals'];

        // Insert data into the goals table

        $sql = "INSERT INTO goals (FixtureID, PlayerID, goals) VALUES (?, ?, ?)";

        $stmt = $conn->prepare($sql);

        // Bind parameters and execute the statement for each scorer

        foreach ($players as $index => $playerID) {

            $stmt->bind\_param("iii", $fixtureID, $playerID, $goals[$index]);

            $stmt->execute();

        }

        // Close statement and database connection

        $stmt->close();

        $conn->close();

        // Redirect to a success page or perform any other actions

        header("Location: add\_scorers.php");

        exit();

    } else {

        // Handle form validation errors

        echo "Error: Please fill in all required fields.";

    }

}

?>

# REGISTER\_ADMINS.HTML

This page is used to register a new administrator

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Register Admin</title>

    <link rel="stylesheet" href="styles.css"> <!-- Link to an external stylesheet (styles.css) for styling -->

</head>

<body>

    <!-- Navigation links -->

    <ul>

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Register Admin</h2>

    <!-- Admin registration form -->

    <form action="register\_admin.php" method="POST"> <!-- Form submission to register\_admin.php using POST method -->

        <label for="username">Username:</label>

        <input type="text" id="username" name="username" required><br><br> <!-- Input field for username with required attribute -->

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" required><br><br> <!-- Input field for password with required attribute -->

        <button type="submit">Register</button> <!-- Submit button to register the admin -->

    </form>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# REGISTER\_ADMIN.PHP

This page is used to add new administrator to the database before redirecting to login page.

<?php

session\_start();

// Check if user is logged in

if (!isset($\_SESSION['username'])) {

    header("Location: login.html");

    exit();

}

include 'connector.php';

// Handle form submission

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    // Check if username already exists

    $sql = "SELECT \* FROM admin\_users WHERE username='$username'";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

        // Username already exists, redirect to registration page with error message

        header("Location: register\_admin.html?error=1");

        exit();

    } else {

        // Encrypt the password

        $hashed\_password = password\_hash($password, PASSWORD\_DEFAULT);

        // Insert new admin user into the database

        $sql = "INSERT INTO admin\_users (username, password) VALUES ('$username', '$hashed\_password')";

        if ($conn->query($sql) === TRUE) {

            // Registration successful, redirect to login page

            header("Location: login.html");

            exit();

        } else {

            // Error in inserting user data, redirect to registration page with error message

            header("Location: register\_admin.html?error=2");

            exit();

        }

    }

}

$conn->close();

?>

# STATISTICS.PHP

This page is used to allow user to select teams tat they want to compare; they then can select the metrics they want to compare and then they can initiate the generation of the bar graph and pie chart. This is then rendered and displayed in a clear and visual way that is optimized using js.

<?php

include 'connector.php';

// Fetch fixtures from the database

$sql = "SELECT FixtureID, HomeTeamID, AwayTeamID, Result FROM Fixtures";

$result = $conn->query($sql);

$fixtures = array();

if ($result->num\_rows > 0) {

    while($row = $result->fetch\_assoc()) {

        if (!empty($row['Result'])) {

            $fixtures[] = $row;

        }

    }

}

// Initialize arrays

$points = array();

$played = array();

$won = array();

$drawn = array();

$lost = array();

$goalsFor = array();

$goalsAgainst = array();

$goalDifference = array();

// Fetch team names from the database

$sql = "SELECT TeamID, TeamName FROM Teams";

$result = $conn->query($sql);

$teams = array();

if ($result->num\_rows > 0) {

    while($row = $result->fetch\_assoc()) {

        $teamID = $row['TeamID'];

        $teamName = $row['TeamName'];

        $teams[$teamID] = $teamName;

        // Initialize team statistics arrays

        $points[$teamName] = 0;

        $played[$teamName] = 0;

        $won[$teamName] = 0;

        $drawn[$teamName] = 0;

        $lost[$teamName] = 0;

        $goalsFor[$teamName] = 0;

        $goalsAgainst[$teamName] = 0;

        $goalDifference[$teamName] = 0;

    }

}

// Process results and calculate points for each team

foreach ($fixtures as $fixture) {

    $result = $fixture['Result'];

    list($home\_score, $away\_score) = explode('-', $result);

    // Update played matches

    $played[$teams[$fixture['HomeTeamID']]]++;

    $played[$teams[$fixture['AwayTeamID']]]++;

    // Update goals for and against

    $goalsFor[$teams[$fixture['HomeTeamID']]] += $home\_score;

    $goalsFor[$teams[$fixture['AwayTeamID']]] += $away\_score;

    $goalsAgainst[$teams[$fixture['HomeTeamID']]] += $away\_score;

    $goalsAgainst[$teams[$fixture['AwayTeamID']]] += $home\_score;

    if ($home\_score > $away\_score) {

        // Home team wins

        $won[$teams[$fixture['HomeTeamID']]]++;

        $lost[$teams[$fixture['AwayTeamID']]]++;

        $points[$teams[$fixture['HomeTeamID']]] += 3;

    } elseif ($home\_score == $away\_score) {

        // Draw

        $drawn[$teams[$fixture['HomeTeamID']]]++;

        $drawn[$teams[$fixture['AwayTeamID']]]++;

        $points[$teams[$fixture['HomeTeamID']]] += 1;

        $points[$teams[$fixture['AwayTeamID']]] += 1;

    } else {

        // Away team wins

        $won[$teams[$fixture['AwayTeamID']]]++;

        $lost[$teams[$fixture['HomeTeamID']]]++;

        $points[$teams[$fixture['AwayTeamID']]] += 3;

    }

}

// Calculate goal difference

foreach ($teams as $teamID => $teamName) {

    $goalDifference[$teamName] = $goalsFor[$teamName] - $goalsAgainst[$teamName];

}

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>League Table Bar Graph</title>

    <link rel="stylesheet" href="styles.css"> <!-- Link to an external stylesheet -->

    <!-- Chart.js library reference -->

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <!-- Explanation: The <script> tag above includes the Chart.js library from a CDN (Content Delivery Network). Chart.js is a popular JavaScript library for creating interactive and responsive charts and graphs in web applications. By including this script tag, we can access the Chart.js library and use its functionalities to create various types of charts and graphs within our web page. Using a CDN ensures that we can easily access the latest version of the library without having to host it on our own server. -->

</head>

<body>

<ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

    <div class="container">

        <form id="teamComparisonForm">

            <label for="team1">Select Team 1:</label>

            <select id="team1" name="team1"></select>

            <label for="team2">Select Team 2:</label>

            <select id="team2" name="team2"></select>

            <label for="metric1">Select Metric 1:</label>

            <select id="metric1" name="metric1">

                <option value="">Select Metric</option>

                <option value="points">Points</option>

                <option value="goalsFor">Goals For</option>

                <option value="goalsAgainst">Goals Against</option>

                <option value="wins">Wins</option>

                <option value="losses">Losses</option>

                <option value="draws">Draws</option>

            </select>

            <label for="metric2">Select Metric 2:</label>

            <select id="metric2" name="metric2">

                <option value="">Select Metric</option>

                <option value="points">Points</option>

                <option value="goalsFor">Goals For</option>

                <option value="goalsAgainst">Goals Against</option>

                <option value="wins">Wins</option>

                <option value="losses">Losses</option>

                <option value="draws">Draws</option>

            </select>

            <label for="metric3">Select Metric 3:</label>

            <select id="metric3" name="metric3">

                <option value="">Select Metric</option>

                <option value="points">Points</option>

                <option value="goalsFor">Goals For</option>

                <option value="goalsAgainst">Goals Against</option>

                <option value="wins">Wins</option>

                <option value="losses">Losses</option>

                <option value="draws">Draws</option>

            </select>

            <button type="button" onclick="updateCharts()">Compare</button>

        </form>

        <canvas id="barChart" width="400" height="200"></canvas>

        <canvas id="pieChart" width="400" height="200"></canvas>

    </div>

    <script>

        // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

        // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

        // If the user is not logged in, hide certain menu items

        if (!isLoggedIn) {

            addResults.style.display = 'none'; // Hide "Add Results" menu item

            addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

            fixtures.style.display = 'none'; // Hide "Fixtures" menu item

            addPlayer.style.display = 'none'; // Hide "Add Player" menu item

            teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

            registerAdmin.style.display = 'none'; // Hide "Register Admin" menu item

            // Redirect user to login page if they try to access restricted pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

            if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

                window.location.href = 'login.html'; // Redirect to login page

            }

        }

        // Retrieve teams and their statistics from PHP

*var* teams = <?php echo json\_encode(array\_values($teams)); ?>; // Get teams

*var* teamStats = {

            "points": <?php echo json\_encode(array\_values($points)); ?>, // Get points data

            "goalsFor": <?php echo json\_encode(array\_values($goalsFor)); ?>, // Get goals for data

            "goalsAgainst": <?php echo json\_encode(array\_values($goalsAgainst)); ?>, // Get goals against data

            "wins": <?php echo json\_encode(array\_values($won)); ?>, // Get wins data

            "losses": <?php echo json\_encode(array\_values($lost)); ?>, // Get losses data

            "draws": <?php echo json\_encode(array\_values($drawn)); ?> // Get draws data

        };

        // Populate team dropdowns once the DOM content is loaded

        document.addEventListener("DOMContentLoaded", *function*() {

*var* team1Select = document.getElementById('team1'); // Get Team 1 dropdown

*var* team2Select = document.getElementById('team2'); // Get Team 2 dropdown

            // Iterate through teams and create options for dropdowns

            teams.forEach(*function*(*team*) {

*var* option1 = document.createElement('option'); // Create option element for Team 1

                option1.value = team; // Set the value of the option

                option1.textContent = team; // Set the text of the option

                team1Select.appendChild(option1); // Append the option to the dropdown

*var* option2 = document.createElement('option'); // Create option element for Team 2

                option2.value = team; // Set the value of the option

                option2.textContent = team; // Set the text of the option

                team2Select.appendChild(option2); // Append the option to the dropdown

            });

        });

        // Function to update charts based on selected teams and metrics

*function* updateCharts() {

*var* team1 = document.getElementById('team1').value; // Get selected team 1

*var* team2 = document.getElementById('team2').value; // Get selected team 2

*var* metric1 = document.getElementById('metric1').value; // Get selected metric 1

*var* metric2 = document.getElementById('metric2').value; // Get selected metric 2

*var* metric3 = document.getElementById('metric3').value; // Get selected metric 3

            // Filter out empty values from selected metrics

*var* selectedMetrics = [metric1, metric2, metric3].filter(Boolean);

            // Retrieve data for selected teams and metrics

*var* team1Data = selectedMetrics.map(*metric* *=>* teamStats[metric][teams.indexOf(team1)]); // Get data for team 1

*var* team2Data = selectedMetrics.map(*metric* *=>* teamStats[metric][teams.indexOf(team2)]); // Get data for team 2

            // Get canvas elements for chart rendering

*var* ctxBar = document.getElementById('barChart').getContext('2d'); // Get context for bar chart

*var* ctxPie = document.getElementById('pieChart').getContext('2d'); // Get context for pie chart

            // Destroy previous charts if they exist

            if (window.myBarChart) {

                window.myBarChart.destroy(); // Destroy bar chart

            }

            if (window.myPieChart) {

                window.myPieChart.destroy(); // Destroy pie chart

            }

            // Create new bar chart

            window.myBarChart = new Chart(ctxBar, {

                type: 'bar',

                data: {

                    labels: selectedMetrics, // Set labels for metrics

                    datasets: [

                        {

                            label: team1, // Set label for team 1

                            data: team1Data, // Set data for team 1

                            backgroundColor: 'rgba(54, 162, 235, 0.5)', // Set color for team 1

                            borderColor: 'rgba(54, 162, 235, 1)', // Set border color for team 1

                            borderWidth: 1 // Set border width for team 1

                        },

                        {

                            label: team2, // Set label for team 2

                            data: team2Data, // Set data for team 2

                            backgroundColor: 'rgba(255, 99, 132, 0.5)', // Set color for team 2

                            borderColor: 'rgba(255, 99, 132, 1)', // Set border color for team 2

                            borderWidth: 1 // Set border width for team 2

                        }

                    ]

                },

                options: {

                    scales: {

                        y: {

                            beginAtZero: true // Ensure y-axis starts at zero

                        }

                    }

                }

            });

            // Create new pie chart with random background colors

*var* pieLabels = selectedMetrics.flatMap(*metric* *=>* [`${team1} - ${metric}`, `${team2} - ${metric}`]); // Set labels for pie chart

*var* pieData = team1Data.concat(team2Data); // Combine data for pie chart

*var* pieBackgroundColors = []; // Initialize array for pie chart colors

            // Generate random colors for pie chart

            for (*let* i = 0; i < pieLabels.length; i++) {

*const* randomColor = `rgba(${Math.floor(Math.random() \* 256)}, ${Math.floor(Math.random() \* 256)}, ${Math.floor(Math.random() \* 256)}, 0.5)`;

                pieBackgroundColors.push(randomColor);

            }

            // Create new pie chart

            window.myPieChart = new Chart(ctxPie, {

                type: 'pie',

                data: {

                    labels: pieLabels, // Set labels for pie chart

                    datasets: [{

                        data: pieData, // Set data for pie chart

                        backgroundColor: pieBackgroundColors, // Set background colors for pie chart

                        borderColor: 'rgba(255, 255, 255, 1)', // Set border color for all sections

                        borderWidth: 1 // Set border width for all sections

                    }]

                },

                options: {

                    responsive: true, // Make chart responsive

                    plugins: {

                        legend: {

                            position: 'top', // Set legend position

                        },

                        tooltip: {

                            callbacks: {

                                label: *function*(*context*) {

                                    return context.label + ': ' + context.raw; // Format tooltip labels

                                }

                            }

                        }

                    }

                }

            });

        }

    </script>

    <footer>

        <!-- Copyright notice -->

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# TEAMS\_FIXTURES.PHP

This page is initiated from the league table page. It displays the fixtures of the team selected.

<?php

include 'connector.php'; // Include the database connector file

// Check if team ID is provided in the URL

if(isset($\_GET['teamID'])) {

    $selectedTeamID = $\_GET['teamID']; // Get the selected team ID from the URL parameter

} else {

    // Redirect to league table page if team ID is not provided

    header("Location: league\_table.php");

    exit();

}

// Fetch fixtures for the selected team from the database

$sql = "SELECT FixtureID, HomeTeamID, AwayTeamID, Result FROM Fixtures WHERE HomeTeamID = $selectedTeamID OR AwayTeamID = $selectedTeamID"; // SQL query to fetch fixtures for the selected team

$result = $conn->query($sql); // Execute the SQL query

$fixtures = array(); // Initialize an empty array to store fixtures

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        if (!empty($row['Result'])) { // Check if the result is not empty

            $fixtures[] = $row; // Add the fixture to the fixtures array

        }

    }

}

// Fetch team names from the database

$teamNames = array(); // Initialize an empty array to store team names

$sql = "SELECT TeamID, TeamName FROM Teams"; // SQL query to fetch team names

$result = $conn->query($sql); // Execute the SQL query

if ($result->num\_rows > 0) { // Check if there are any rows returned by the query

    while($row = $result->fetch\_assoc()) { // Loop through each row of the result set

        $teamNames[$row['TeamID']] = $row['TeamName']; // Store team name in

the array with TeamID as the key

    }

}

// Close the database connection

$conn->close(); // Close the database connection after fetching data

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Team Fixtures</title>

    <link rel="stylesheet" href="styles.css"> <!-- Link to an external stylesheet -->

</head>

<body>

<ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Team Fixtures</h2> <!-- Heading for the team fixtures -->

    <div class="table-container">

    <h3>Fixtures for <?php echo $teamNames[$selectedTeamID]; ?></h3> <!-- Display selected team name -->

    <table border="1"> <!-- Table to display the team fixtures -->

        <tr>

            <th>Home Team</th>

            <th>Result</th>

            <th>Away Team</th>

        </tr>

        <?php foreach ($fixtures as $fixture): ?> <!-- Loop through fixtures -->

            <tr class="clickable-row" data-href="fixture\_outcome.php?fixture\_id=<?php echo $fixture['FixtureID']; ?>">

                <td><?php echo $teamNames[$fixture['HomeTeamID']]; ?></td> <!-- Display Home Team Name -->

                <td><?php echo $fixture['Result']; ?></td> <!-- Display Result -->

                <td><?php echo $teamNames[$fixture['AwayTeamID']]; ?></td> <!-- Display Away Team Name -->

            </tr>

        <?php endforeach; ?> <!-- End of fixture loop -->

    </table>

        </div>

        <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

<script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

<!-- JavaScript to make rows clickable -->

<script>

    document.addEventListener("DOMContentLoaded", *function*() {

*var* rows = document.querySelectorAll(".clickable-row"); // Select all rows with the class "clickable-row"

        rows.forEach(*function*(*row*) {

            row.addEventListener("click", *function*() {

*var* href = row.getAttribute("data-href"); // Get the URL from the "data-href" attribute

                window.location.href = href; // Navigate to the URL when the row is clicked

            });

        });

    });

</script>

# TEAMS\_INPUT.PHP

This page can be used to input new teams into the database.

<?php

    include 'connector.php';

    session\_start(); // Start the session to manage user session data

// Check if the user is not logged in (session variable not set)

if (!isset($\_SESSION['username'])) {

    header("Location: login.html"); // Redirect to the login page

    exit(); // Terminate the script execution

}

    if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

        $team\_name = $\_POST['team\_name'];

        $team\_logo = $\_POST['team\_logo']; // Assuming team logo is stored as a URL

        // Insert new team into the database

        $sql = "INSERT INTO Teams (TeamName, TeamLogo) VALUES ('$team\_name', '$team\_logo')";

        if ($conn->query($sql) === TRUE) {

            // Redirect to teams\_input.html with success message

            header("Location: teams\_input.php?status=success");

            exit();

        } else {

            // Redirect to teams\_input.html with error message

            header("Location: teams\_input.php?status=error");

            exit();

        }

    }

    $conn->close();

    ?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Add New Team</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <ul>

        <!-- List of navigation links to other pages -->

        <li><a href="league\_table.php">League Table</a></li>

        <li><a href="add\_results.php">Add Results</a></li>

        <li><a href="add\_scorers.php">Add Scorers</a></li>

        <li><a href="top\_scorers.php">Top Scorers</a></li>

        <li><a href="statistics.php">Statistics</a></li>

        <li><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li><a href="fixtures.php">Fixtures</a></li>

        <li><a href="add\_player.php">Add Player</a></li>

        <li><a href="teams\_input.php">Teams Input</a></li>

        <li><a href="register\_admin.html">Register admin</a></li>

    </ul>

    <h2>Add New Team</h2>

    <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

        <label for="team\_name">Team Name:</label>

        <input type="text" id="team\_name" name="team\_name" required><br><br>

        <label for="team\_logo">Team Logo:</label>

        <input type="text" id="team\_logo" name="team\_logo"><br><br>

        <button type="submit">Add Team</button>

    </form>

    <footer>

        <p>&copy; 2024 EPL. All rights reserved.</p>

    </footer>

</body>

</html>

# TOP\_SCORERS.PHP

This page is used to render the top scorer page. The top scorers can be filtered according to their different teams. You can select on a player to see in which fixtures they scored in.

<?php

include 'connector.php';

// Fetch goals scored by each player from the database

$sql = "SELECT Players.PlayerID, Players.PlayerName, Teams.TeamName, COUNT(Goals.PlayerID) AS GoalsScored

            FROM Players

            INNER JOIN Teams ON Players.TeamID = Teams.TeamID

            LEFT JOIN Goals ON Players.PlayerID = Goals.PlayerID

            GROUP BY Players.PlayerID

            ";

$result = $conn->query($sql);

$goalscorers = array();

if ($result->num\_rows > 0) {

    while($row = $result->fetch\_assoc()) {

        $goalscorers[] = $row;

    }

}

// Fetch all teams for the team filter

$sql = "SELECT TeamName FROM Teams";

$result = $conn->query($sql);

$teams = array(); // Initialize an empty array to hold the team names

// Check if the result has more than 0 rows

if ($result->num\_rows > 0) { // If there are rows in the result set

    // Loop through each row in the result set

    while($row = $result->fetch\_assoc()) { // Fetch a row as an associative array

        // Add the value of 'TeamName' from the current row to the $teams array

        $teams[] = $row['TeamName'];

    }

}

// Handle team filter

if(isset($\_POST['team\_filter'])) {

    $selected\_team = $\_POST['team\_filter'];

    // If a team is selected, filter goalscorers by that team

    if ($selected\_team != 'All') {

        $filtered\_goalscorers = array\_filter($goalscorers, *function*($goalscorer) use ($selected\_team) {

            return $goalscorer['TeamName'] == $selected\_team;

        });

    } else {

        $filtered\_goalscorers = $goalscorers;

    }

} else {

    // If no team filter is provided, display all goalscorers

    $filtered\_goalscorers = $goalscorers;

}

// Sort the filtered goalscorers array by the 'Goals Scored' column in descending order

usort($filtered\_goalscorers, *function*($a, $b) {

    return $b['GoalsScored'] - $a['GoalsScored'];

});

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Top Scorers</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

<ul>

        <!-- List of navigation links to other pages -->

        <li id="league\_table"><a href="league\_table.php">League Table</a></li>

        <li id="add\_results"><a href="add\_results.php">Add Results</a></li>

        <li id="add\_scorers"><a href="add\_scorers.php">Add Scorers</a></li>

        <li id="top\_scorers"><a href="top\_scorers.php">Top Scorers</a></li>

        <li id="statistics"><a href="statistics.php">Statistics</a></li>

        <li id="display\_fixtures"><a href="display\_fixtures.php">Display Fixtures</a></li>

        <li id="fixtures"><a href="fixtures.php">Fixtures</a></li>

        <li id="add\_player"><a href="add\_player.php">Add Player</a></li>

        <li id="teams\_input"><a href="teams\_input.php">Teams Input</a></li>

        <li id="register\_admin"><a href="register\_admin.html">Register admin</a></li>

        <li id="Login"><a href="Login.html">Login</a></li>

    </ul>

    <script>

    // Check if the cookie named "username" exists

*const* isLoggedIn = document.cookie.split(';').some((*item*) *=>* item.trim().startsWith('username='));

    // Get the list items corresponding to the menu items you want to hide/deny access to

*const* addResults = document.getElementById('add\_results');

*const* addScorers = document.getElementById('add\_scorers');

*const* fixtures = document.getElementById('fixtures');

*const* addPlayer = document.getElementById('add\_player');

*const* teamsInput = document.getElementById('teams\_input');

*const* registerAdmin = document.getElementById('register\_admin');

    if (!isLoggedIn) {

        // User is not logged in, hide menu items and deny access to pages

        addResults.style.display = 'none'; // Hide "Add Results" menu item

        addScorers.style.display = 'none'; // Hide "Add Scorers" menu item

        fixtures.style.display = 'none'; // Hide "Fixtures" menu item

        addPlayer.style.display = 'none'; // Hide "Add Player" menu item

        teamsInput.style.display = 'none'; // Hide "Teams Input" menu item

        registerAdmin.style.display = 'none'; // Hide "Register admin" menu item

        // Redirect user to login page if they try to access certain pages directly

*const* restrictedPages = ['add\_results.php', 'add\_scorers.php', 'fixtures.php', 'add\_player.php', 'teams\_input.php', 'register\_admin.html'];

        if (restrictedPages.includes(window.location.pathname.split('/').pop())) {

            window.location.href = 'login.html'; // Redirect to login page

        }

    }

</script>

    <h2>Top Scorers</h2> <!-- Header for the section -->

    <div class="table-container"> <!-- Container for the table -->

    <form action="top\_scorers.php" method="POST"> <!-- Form to filter top scorers by team -->

        <label for="team\_filter">Filter by Team:</label> <!-- Label for the team filter dropdown -->

        <select id="team\_filter" name="team\_filter"> <!-- Dropdown for selecting a team to filter -->

            <option value="All">All</option> <!-- Option to show all teams -->

            <?php foreach ($teams as $team): ?> <!-- Loop through each team in the $teams array -->

                <option value="<?php echo $team; ?>"><?php echo $team; ?></option> <!-- Option for each team -->

            <?php endforeach; ?>

        </select>

        <button type="submit">Apply Filter</button> <!-- Button to submit the form -->

    </form>

    <table id="top-scorers-table"> <!-- Table to display top scorers -->

        <thead>

            <tr>

                <th>Player Name</th> <!-- Table header for player names -->

                <th>Team</th> <!-- Table header for team names -->

                <th>Goals Scored</th> <!-- Table header for goals scored -->

            </tr>

        </thead>

        <tbody>

            <?php foreach ($filtered\_goalscorers as $goalscorer): ?> <!-- Loop through each filtered goalscorer -->

                <?php if ($goalscorer['GoalsScored'] > 0): ?> <!-- Check if the player has scored more than 0 goals -->

                    <tr class="clickable-row" data-href="player\_fixtures.php?player\_id=<?php echo $goalscorer['PlayerID']; ?>"> <!-- Table row for each goalscorer with clickable link -->

                        <td><?php echo $goalscorer['PlayerName']; ?></td> <!-- Display player name -->

                        <td><?php echo $goalscorer['TeamName']; ?></td> <!-- Display team name -->

                        <td><?php echo $goalscorer['GoalsScored']; ?></td> <!-- Display goals scored -->

                    </tr>

                <?php endif; ?>

            <?php endforeach; ?>

        </tbody>

    </table>

                </div>

                <footer> <!-- Footer section -->

        <p>&copy; 2024 EPL. All rights reserved.</p> <!-- Copyright notice -->

    </footer>

    <script>

        document.addEventListener("DOMContentLoaded", *function*() { // Ensure the script runs after the DOM is fully loaded

*var* table = document.getElementById("top-scorers-table"); // Get the table element by its ID

*var* rows = table.getElementsByTagName("tr"); // Get all rows in the table

            for (*var* i = 0; i < rows.length; i++) { // Loop through each row

*var* row = rows[i];

                row.addEventListener("click", *function*() { // Add click event listener to each row

*var* href = this.dataset.href; // Get the data-href attribute of the clicked row

                    if (href) { // If there is a href value

                        window.location.href = href; // Navigate to the URL specified in data-href

                    }

                });

            }

        });

    </script>

</body>  
</html>

# STYLES.CSS

This is the css used to render my page.

/\* Set the font family and background color for the entire page \*/

body {

*font-family*: Arial, sans-serif;

*background-color*: #f2f2f2;

*margin*: 20px; /\* Add margin to the edges of the page \*/

}

/\* Center-align heading \*/

h2 {

*text-align*: center;

*margin-top*: 20px; /\* Add margin space above the heading \*/

}

/\* Style the unordered list (menu) \*/

ul {

*list-style-type*: none; /\* Remove default list bullets \*/

*margin*: 0;

*padding*: 0;

*text-align*: center; /\* Center-align the list items \*/

*color*: white; /\* Set text color to white \*/

*background-color*: #090909; /\* Background color for the list menu \*/

*padding*: 10px; /\* Add padding to the list menu \*/

}

/\* Style the list items \*/

ul li {

*display*: inline; /\* Display list items inline \*/

*margin*: 0 10px; /\* Add margin space around each list item \*/

}

/\* Style the table \*/

table {

*width*: 100%; /\* Make the table occupy 100% of its container \*/

*border-collapse*: collapse; /\* Collapse table borders \*/

*margin-top*: 20px; /\* Add margin space above the table \*/

*border*: 2px solid #333; /\* Add border to the table \*/

}

/\* Style table header and data cells \*/

table th, table td {

*padding*: 10px; /\* Add padding to table cells \*/

*text-align*: center; /\* Center-align text within cells \*/

*border*: 1px solid #333; /\* Add border to table cells \*/

}

/\* Style table header \*/

table th {

*background-color*: #333; /\* Background color for table headers \*/

*color*: #fff; /\* Text color for table headers \*/

}

/\* Alternate row colors for table \*/

table tr:nth-child(even) {

*background-color*: #f2f2f2; /\* Background color for even rows \*/

}

/\* Highlight row on hover \*/

table tr:hover {

*background-color*: #ddd; /\* Background color for hovered rows \*/

}

/\* Style hyperlinks \*/

a {

*text-decoration*: none; /\* Remove underline from hyperlinks \*/

*color*: #fff; /\* Set text color for hyperlinks \*/

}

/\* Style hyperlinks on hover \*/

a:hover {

*background-color*: #3e3c3c; /\* Background color for hyperlinks on hover \*/

}

/\* Left-align team name cells \*/

table td:nth-child(2) {

*text-align*: left; /\* Left-align text within team name cells \*/

}

/\* Style the form \*/

form {

*text-align*: center; /\* Center-align form elements \*/

*margin-top*: 20px; /\* Add margin space above the form \*/

}

/\* Style form labels \*/

form label {

*display*: block; /\* Display labels as block elements \*/

*margin-bottom*: 10px; /\* Add margin space below each label \*/

}

/\* Style the submit button \*/

form button[type="submit"] {

*background-color*: #4CAF50; /\* Set button background color \*/

*color*: white; /\* Set button text color \*/

*padding*: 10px 20px; /\* Add padding to button \*/

*border*: none; /\* Remove button border \*/

*cursor*: pointer; /\* Set cursor to pointer on hover \*/

*border-radius*: 4px; /\* Add border radius to button \*/

}

/\* Style the submit button on hover \*/

form button[type="submit"]:hover {

*background-color*: #45a049; /\* Darker background color on hover \*/

}

/\* Style form input fields \*/

form select,

form input[type="date"], input[type="text"],input[type="password"],input[type="number"] {

*width*: 50%; /\* Set input fields to occupy 50% of the width \*/

*padding*: 8px; /\* Add padding to input fields \*/

*margin-bottom*: 10px; /\* Add margin space below each input field \*/

*box-sizing*: border-box; /\* Include padding and border in the element's total width and height \*/

}

/\* Style the container for scorer entries \*/

#scorersContainer {

*margin-bottom*: 20px; /\* Add margin space below the scorers container \*/

}

/\* Style scorer entry fields \*/

.scorerEntry {

*margin-bottom*: 10px; /\* Add margin space below each scorer entry \*/

}

/\* Style the add scorer button \*/

#addScorerButton {

*margin-bottom*: 10px; /\* Add margin space below the add scorer button \*/

}

label {

*display*: block; /\* Display labels as block elements \*/

*margin-bottom*: 10px; /\* Add margin space below labels \*/

}

/\* Style the dashboard grid \*/

.dashboard-grid {

*display*: grid;

*grid-template-columns*: repeat(auto-fit, minmax(200px, 1fr));

*grid-auto-rows*: minmax(100px, auto); /\* Automatically adjust row height \*/

*gap*: 20px;

*list-style-type*: none;

*text-align*: center;

*padding*: 0;

*margin*: 0; /\* Remove default margin \*/

*height*: 100%; /\* Make grid take up full height of body \*/

}

/\* Style the dashboard grid items \*/

.dashboard-grid li { /\* Selects the <li> elements inside an element with the class "dashboard-grid" \*/

*background-color*: #4CAF50; /\* Sets the background color to a shade of green \*/

*padding*: 20px; /\* Adds 20 pixels of padding inside each grid item \*/

*border-radius*: 10px; /\* Rounds the corners with a radius of 10 pixels \*/

*text-align*: center; /\* Centers the text horizontally \*/

*box-shadow*: 0 4px 6px rgba(0, 0, 0, 0.1); /\* Adds a subtle shadow for a raised effect \*/

}

/\* Style the links inside grid items \*/

.dashboard-grid li a { /\* Selects the <a> elements inside <li> elements that are inside an element with the class "dashboard-grid" \*/

*text-decoration*: none; /\* Removes the underline from the links \*/

*color*: #333; /\* Sets the text color to a dark gray \*/

*text-align*: center; /\* Centers the text horizontally \*/

*font-size*: 18px; /\* Sets the font size to 18 pixels \*/

*font-weight*: bold; /\* Makes the text bold \*/

}

.table-container {

*text-align*: center; /\* Center-align the table \*/

*max-width*: 70%; /\* Limit maximum width of the table container \*/

*margin*: 0 auto; /\* Center the table container horizontally \*/

}

/\* Container Styles \*/

.container {

*max-width*: 1200px; /\* Maximum width of the container \*/

*margin*: 20px auto; /\* Center the container with top and bottom margins \*/

*padding*: 20px; /\* Add padding inside the container \*/

*background-color*: white; /\* White background for the container \*/

*box-shadow*: 0 0 10px rgba(0, 0, 0, 0.1); /\* Slight shadow for elevation \*/

*border-radius*: 8px; /\* Rounded corners \*/

}

/\* Form Styles \*/

.container form {

*display*: flex; /\* Use flexbox for form layout \*/

*flex-direction*: row; /\* Arrange form elements horizontally \*/

*flex-wrap*: wrap; /\* Allow flex items to wrap onto multiple lines \*/

*gap*: 20px; /\* Add space between flex items \*/

}

.container form label {

*flex*: 0 0 auto; /\* Label takes content width \*/

*width*: 150px; /\* Fixed width for labels \*/

*font-weight*: bold; /\* Bold text for labels \*/

}

.container form select,

.container form button {

*flex*: 1 1 auto; /\* Select and button elements take remaining space \*/

*padding*: 8px; /\* Add padding inside the elements \*/

*margin-top*: 5px; /\* Margin on top of elements \*/

*margin-bottom*: 20px; /\* Margin below elements \*/

*font-size*: 14px; /\* Reduced font size for select and button \*/

*border*: 1px solid #ccc; /\* Light gray border \*/

*border-radius*: 4px; /\* Rounded corners \*/

*width*: 100px; /\* Reduced width for select elements \*/

}

.container form button {

*background-color*: #4CAF50; /\* Green background for buttons \*/

*color*: white; /\* White text color \*/

*cursor*: pointer; /\* Pointer cursor on hover \*/

*border*: none; /\* Remove border \*/

}

.container form button:hover {

*background-color*: #45a049; /\* Darker green on hover \*/

}

/\* Chart Styles \*/

canvas {

*margin*: 20px 0; /\* Margin around canvas elements \*/

*max-width*: 100%; /\* Ensure the chart fits within its container \*/

*height*: auto; /\* Maintain aspect ratio \*/

}

#barChart,

#pieChart {

*max-width*: 400px; /\* Set a maximum width for the charts \*/

*height*: auto; /\* Maintain aspect ratio \*/

*margin*: 20px auto; /\* Center the charts horizontally with margin \*/

}

/\* Footer styles: Styling for the footer section of the page \*/

footer {

*background-color*: #333; /\* Background color for the footer \*/

*color*: #fff; /\* Text color for the footer \*/

*text-align*: center; /\* Center-align the content of the footer \*/

*padding*: 0px 0; /\* Padding top and bottom of the footer \*/

*position*: fixed; /\* Fix the footer at the bottom of the viewport \*/

*bottom*: 0; /\* Align the footer at the bottom \*/

*width*: 100%; /\* Full width of the viewport \*/

*border-top*: 10px solid #fff; /\* Solid border at the top of the footer \*/

*box-sizing*: border-box; /\* Include padding and border in the element's total width and height \*/

}

REFERENCES:

<!-- Chart.js library reference -->

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>